AMENDMENT TO THE CLAIMS

1. (Currently Amended) A method for detecting and classifying block edges from DCT (Discrete Cosine Transform)-compressed images~~, which is for detecting the edge of each block from DCT-compressed images and classifying an edge direction component of each block~~, the method comprising:

(a) extracting DCT coefficients by N × N blocks constituting ~~the~~ a compressed image;

(a1) forming a plurality of regions in each block, each region defining a direction; and

(b) applying an arithmetic operation defined for each direction ~~component~~ to the DCT coefficients obtained in (a), and comparing ~~the~~ results of the arithmetic operations among the directions of a corresponding block to determine ~~the~~ an edge direction component for the corresponding block, the arithmetic operations using an averaged luminance value of each region represented in terms of the DCT coefficients.

2. (Original) The method as claimed in claim 1, wherein the arithmetic operation of (b) comprises a combination of H, V, and D that are the weighted sums of the extracted DCT coefficients as given by the following equations:

$$H = \sum_{\substack{n=1 \\ (odd)}}^{N-1} w(0,n) X(0,n)$$

$$V = \sum_{\substack{n=1 \\ (odd)}}^{N-1} w(m,0) X(m,0)$$

$$D=\sum_{\substack{n=1\\(odd)}}^{N-1} w(n,n)X(n,n)+\sum_{\substack{m=1\\(odd)}}^{N-1}\sum_{\substack{n=1\\(odd)}}^{m-1} w(m,n)\big(X(m,n)+X(n,m)\big)$$

where m and n represent spatial frequency components in vertical and horizontal directions, respectively, and satisfy $0 \le m, n \le N$.

3. (Currently Amended) The method as claimed in claim 1, wherein (a) comprises extracting Y(m, n) = w(m, n)X(m, n) that is a multiplication of the DCT coefficient X(m, n) by a quantitative numeric, rather than simply extracting the DCT coefficient,

wherein (b) of performing each arithmetic operation comprises determining H, V, and D according to the following equations to construct an arithmetic operation for each ~~edge~~ direction using a combination of H, V, and D:

$$H=\sum_{\substack{n=1\\(odd)}}^{N-1} Y(0,n),\quad V=\sum_{\substack{n=1\\(odd)}}^{N-1} Y(m,0),\quad D=\sum_{\substack{m=1\\(odd)}}^{N-1}\sum_{\substack{n=1\\(odd)}}^{N-1} Y(m,n)\underline{.}$$

4. (Currently Amended) The method as claimed in claim 2, wherein (b) of calculating H, V, and D comprises performing summations up to upper limits of the "m" and "n" set to L ($1 \le L \le N-1$) rather than N-1 to construct the arithmetic operation for each ~~edge~~ direction ~~component~~.

5. (Currently Amended) The method as claimed in claim 3, wherein (b) of calculating H, V, and D comprises performing summations up to upper limits of the

"m" and "n" set to L ($1 \le L \le N-1$) rather than N-1 to construct the arithmetic operation for each ~~edge~~ direction ~~component~~.

6. (Original) The method as claimed in claim 4, wherein (b) comprises setting L = 1, and computing H, V, and D with $w(0,1) = w(1,0) = w(1,1) \approx 0.5\alpha$.

7. (Currently Amended) The method as claimed in claim 2, wherein w(m, n) of (b) is calculated according to the following equation:

$$w(m,n) = \alpha e(m)e(n)C_{16}^{4m}C_{16}^{2m}C_{16}^{m}C_{16}^{4n}C_{16}^{2n}C_{16}^{n}$$

wherein $C_{2N}^{\tau} = \cos\frac{\pi\tau}{2N}$; $e(\tau) = \begin{cases} 1/\sqrt{2}, & if \quad \tau = 0 \\ 1, & elsewhere \end{cases}$; and α is a multiplicative constant ~~determined in consideration of different variants of the DCT equation~~.

8. (Currently Amended) The method as claimed in claim 3, wherein w(m, n) of (b) is calculated according to the following equation:

$$w(m,n) = \alpha e(m)e(n)C_{16}^{4m}C_{16}^{2m}C_{16}^{m}C_{16}^{4n}C_{16}^{2n}C_{16}^{n}$$

wherein $C_{2N}^{\tau} = \cos\frac{\pi\tau}{2N}$; $e(\tau) = \begin{cases} 1/\sqrt{2}, & if \quad \tau = 0 \\ 1, & elsewhere \end{cases}$; and α is a multiplicative constant ~~determined in consideration of different variants of the DCT equation~~.

9. (Currently Amended) The method as claimed in claim 2, wherein (b) comprises classifying each ~~edge~~ direction ~~component~~ into NE (No Edge), 0 radian, π/4

radian, $\pi/2$ radian, and $3\pi/4$ radian, computing ~~measuring~~ the arithmetic operation of each direction~~al~~ ~~component~~ using the H, V, and D determined in claim 2 ~~or 3~~ according to the following equations, and ~~classifying~~ determining the edge direction component of each block ~~edge~~ as the ~~the~~ one ~~whose measure is the~~having the greatest value among ~~others~~the directions of the corresponding block:

δ_{NE} (set by a user)

$$\delta_0 = 2|V|$$

$$\delta_{\pi/4} = \frac{4}{3}\max\{|H+V+D|,|H+V-D|\}$$

$$\delta_{\pi/2} = 2|H|$$

$$\delta_{3\pi/4} = \frac{4}{3}\max\{|H-V+D|,|H-V-D|\}.$$

10. (Currently Amended) The method as claimed in claim 3, wherein (b) comprises classifying each ~~edge~~ direction ~~component~~ into NE (No Edge), 0 radian, $\pi/4$ radian, $\pi/2$ radian, and $3\pi/4$ radian, computing ~~measuring~~ the arithmetic operation of each direction~~al~~ ~~component~~ using the H, V, and D determined in claim ~~2 or~~ 3 according to the following equations, and ~~classifying~~ determining the edge direction component of each block ~~edge~~ as the ~~the~~ one ~~whose measure is~~having the greatest value among ~~others~~the directions of the corresponding block:

δ_{NE} (set by a user)

$$\delta_0 = 2|V|$$

$$\delta_{\pi/4} = \frac{4}{3}\max\{|H + V + D|, |H + V - D|\}$$

$$\delta_{\pi/2} = 2|H|$$

$$\delta_{3\pi/4} = \frac{4}{3}\max\{|H - V + D|, |H - V - D|\}.$$